Hardware Results of Chaotic Time Series Prediction using Cascade Error Projection Algorithm

Tuan A. Duong and Brent R. Blaes
Center for Integrated Space Microsystems
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

Abstract:
This paper briefly describes an on-line hardware system based on a 16x8x4 cascading neural network architecture. The chip was fabricated through MOSIS on a 0.35um CMOS-bulk process and successfully tested for functionality. This chip is used to evaluate a neural network solution in VLSI hardware using chaotic time series containing very high order correlation data that never repeats itself. Chaotic time series is an excellent tool for evaluating the robustness of prediction techniques. We also describe the technique used to perform on-line learning using an algorithm that easily maps to hardware called Cascade Error Projection (CEP). This technique consists of two steps: 1) down loading into the chip a set of 8-bit quantized weights, and 2) on-line adjustment (re-learning) of the new weight set to compensate for mismatching induced by quantization and transistor mismatch on the chip.
The results will be compared with the software-based technique (8-bit or 64-bit quantization weight) using CEP.
This work, based on noise-prediction and subtraction techniques, will be developed for on-chip learning and applied to systems where adaptive noise cancellation is required to meet system requirements. It is currently being applied to reducing noise and enhancing the performance of a MEMS-based micro gyro system that is currently under development. Application of this work to large mixed-signal chip designs is planned for Systems-On-A-Chip (SOAC) applications for the Center for Integrated Microsystems (CISM) program at Jet Propulsion Laboratory.

I. Introduction:

On-chip noise is a major factor that needs to be dealt with when designing highly integrated mixed-signal systems on a chip. Self-induced on-chip noise, which degrades system performance, is very difficult to deal with. This is especially true for space-based applications such navigation-grade MEMS-based micro gyro systems requiring bias drift stabilities as low as 0.01^0/hr. To meet this demand, on-chip noise reduction is a necessity. One way to reduce noise is through prediction and subtraction (noise cancellation). However, noise prediction is a challenging task and sometimes cannot be done, e.g. random noise cannot be predicted. Noise that has very weak correlation with its past is studied in this paper. We use a chaotic time series prediction algorithm that is applicable to a noise source that never repeat itself and contains a very weak correlation with the past (the first and seventeenth delay units) [1].

Neural networks have been successfully used for prediction [1-5] and are a good candidate for this application. However, most of the work in this area has utilized software based simulation networks that may not be suitable for SOAC hardware implementation. Additionally, some reports have investigated neural network learning under limited bit weight quantization [6-8] and focused on prediction problems [2].

In this paper, our study will focus on a technique to map the software solution into a hardware system and to

relearn new weight sets to compensate for element mismatching in the hardware. Accordingly, we will present the system description in section II, hardware results from a downloaded software weight set in section III, a technique for on-line learning in section VI, and conclusions in section V.

II. System Description:

In previous studies [2,8], CEP has demonstrated tolerance to less bit quantization for learning, than for other learning techniques that require more bits [6-7]. In this study, we designed, fabricated, and tested a 16x8x4 CEP neural chip. The chip was fabricated through TSMC's 0.35um CMOS-bulk technology via MOSIS. The system architecture is shown below in Figure 1.

All components such as 8-bit DAC inputs and synaptic weights can be programmed through a row and column address. In addition, neuron sigmoidal slope can also be varied programmable input Igain (Igain is low, the neuron slope is stiff and vice versa).

Input array:

The input array has 16 (i_1-i_{16}) 8-bit DACs as shown in Figure 1. The detail design of this DAC can be found in [9]. The output of each DAC consists of two currents of value Iin and 16*Iin. The output pairs from the 8-bit DACs are input to synapse to perform a two quadrant multiplication (input is positive 8-bit while synapse is +/- 7-bit weight). The outputs of the multipliers are summed vertically through a wire and connected to the input of a neuron to get a sigmoidal mapping.

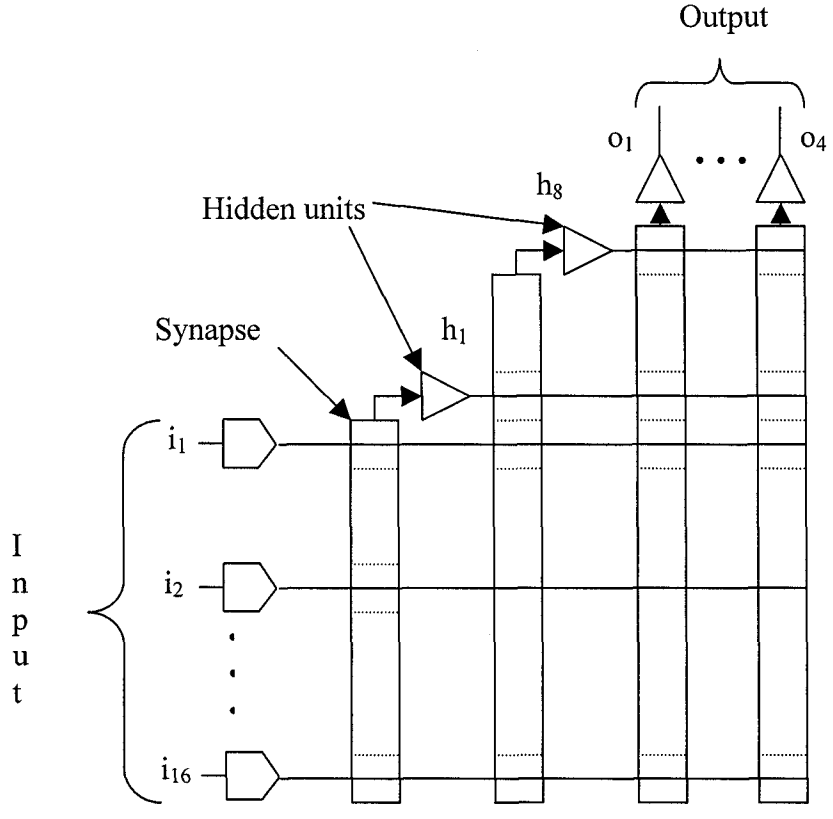

Figure 1. Cascade error projection neural architecture implemented in TSMC's 0.35um CMOS-bulk technology.

Hidden units:

There are 8 hidden units in this system (Figure 1). The weight set in a hidden unit is incrementally one synapse from the previous one (hidden unit 2 contains 17 synapses while hidden unit 1 is only contains 16 synapses) and their inputs are also obtained from the output of the previous hidden unit.

The operation of hidden units is: 1) inner product of input plus hidden units (if there are previous hidden units to input of current hidden units) and weight vectors; 2) the output results of inner product vectors are summed and mapped to sigmoidal function. The slope of sigmoidal function can be adjusted by varying the current Igain.

Output units:

There are four output units (Figure 1) and each output is mapped through a sigmoidal function by inner product of inputs and 8 hidden units with 24 synaptic weights. The slope of the sigmoidal function can be varied the same as that of the hidden units.

III. Weight Down Loading:

In this study, we only used four input units as x_i, x_{i+1}, x_{i+2}, and x_{i+3} and one output is x_{i+4}.

The chaotic time series prediction problem has been studied through simulation of the CEP algorithm with floating point (32-bit), 4- to 8-bit weight quantization and the results are reported in [2].

The weight set that is obtained by software simulation using a floating-point machine is down loaded to our hardware system as shown above. The results obtained by hardware are shown below:

The error curve of the original data and hardware results (from down loading

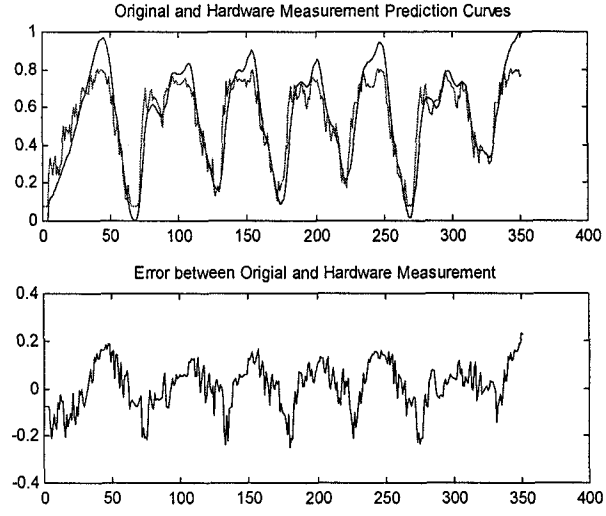

Figure 2. output results from hardware from downloading weight set. The top plots consist of original and hardware results; and the bottom plot is the error between the top plots.

weight) is not sufficient (Figure 2).
The discrepancy of results between hardware and original data (Figure 2) may come from mismatching transistor, sigmoidal curve, noise from system etc... To compensate for the mismatching, we utilize a re-learning phase, which will add a new hidden unit to perform on-line learning based on the current error surface.

IV. On-Line Learning:

When the results from the weight downloading approach are not satisfied for the application, relearning in hardware (on-line learning) is the next step utilized for compensating the

mismatching in hardware. The Cascade Error Projection (CEP) learning technique is used. The advantages of CEP is simple to learn and friendlier to hardware limitation such as limited bit weight quantization. For the CEP on line learning, there consists of two steps:

a) Perceptron learning:
In this session, the error surface (the differences between target and actual output data) is projected to a newly added hidden unit as a new target in which the new energy function is formed:

$$E(n+1)=\sum_{p=1}^{P}\{f_h^p(n+1)-\frac{1}{m}\sum_{o=1}^{m}(t_o^p-o_o^p)\}^2$$

With n - number of previous hidden unit, p- index of pattern number, P-total pattern, o-index of output number, m-total output required.
The new weight can be updated by:

$$\Delta w_{ih}^p(n+1)=-\eta\frac{\partial E(n+1)}{\partial w_{ih}^p(n+1)}$$

In this hardware learning process, the derivative of the sigmoidal transfer function of hidden unit n+1 is required. To obtain a derivative in hardware, we use a bias as input while the weight is changed with +/- 5. The respected output of hidden unit n+1 while changing weights are used to calculate the derivative as below:

$$f'(x)=\alpha*(f(x+5)-f(x-5))$$

b) Calculating weight:
After 500 learning iterations is complete, the weight set between new hidden unit n+1 and output units can be calculated as follow:

$$w_{ho}=\frac{\sum_{p=1}^{P}\varepsilon_o^p f'^p_o f_h^p(n+1)}{\sum_{p=1}^{P}[f'^p_o f_h^p(n+1)]^2}$$

Error $\varepsilon_o^p=t_o^p-o_o^p(n)$; $f'^p_o(n)=f'^p_o$ denotes the output transfer function derivative with respect to its input.
$f_h^p(n+1)$ - the transfer function of hidden unit $n+1$. This procedure is repeated with three newly added hidden units consecutively, the output results were obtained in Figure 3.

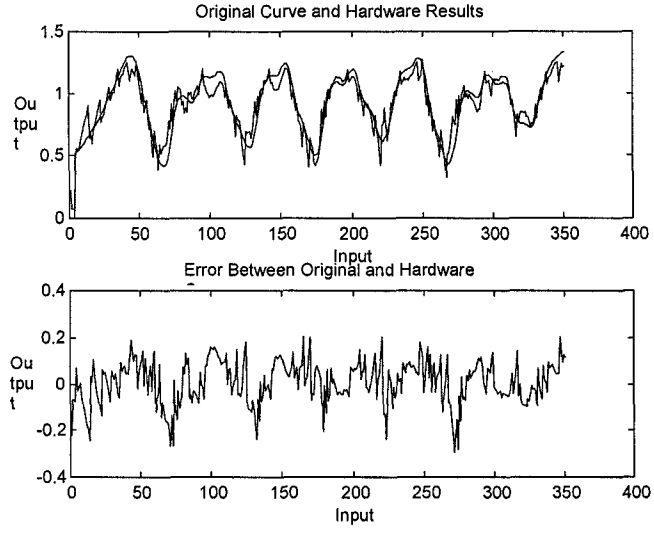

Figure 3. output results from hardware after on-line training. The top plots consist of original and hardware results after re-learning; and the bottom plot is the error between the top plots.

The results show some improvements when on-line learning is applied. It is believed that when more hidden units are added, the performance is better.

V. Conclusion:

In this paper, we demonstrated that online learning in hardware using CEP is a sufficient technique to compensate for hardware mismatching due to processing variation and limited bit quantization to solve time series prediction problems. From this benchmark problem, it can be suitably applied to solve noise prediction when correlated noise exists. This study is still a pre-final step to the noise reduction problem.

For the future study, on-chip learning will be the final goal in solving practical problems involving noise that can vary in time.

Acknowledgments:

The research described herein was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly sponsored by the Ballistic Missile Defense Organization (BMDO), and the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs T. Daud, and A. Thakoor for useful discussions.

References:

1. E. Littmann and H. Ritter, "*Learning and Generalization in Cascade Network Architectures*," Neural Computation, Vol. 8, NO. 7, pp. 1521-1540, Oct. 1996.

2. T.A. Duong and T. Daud, "*Cascade Error Projection with low bit weight quantization for high order correlation data*", IJCNN of IEEE in Washington D.C. Vol. 3, pp. 1600-1603, 1999.

3. Z. Trajanoski and P. Wach, "*Neural Predictive Controller for Insulin Delivery Using the Subcutaneous Route*", IEEE Tran. On Biomedical Engineering, Vol. 45, NO. 9, pp. 1122-1127, Sep. 1998.

4. A.A. Ilumoka, "*Efficient prediction of interconnect crosstalk using neural networks*", Proceedings. 12th IEEE International Conference on Tools with Artificial Intelligence ICTAI 2000, pp. 122 –125, 2000.

5. T. Matsuura, T. Hiei, H. Itoh, K. Torikoshi. "*Active noise control by using prediction of time series data with a neural network*" IEEE International Conference on Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century, Vol. 3, pp. 2070 - 2075, 1995.

6. P. W. Hollis, J.S. Harper, and J.J. Paulos, "*The effects of Precision Constraints in a Backpropagation learning Network*," *Neural Computation*, vol. 2, pp. 363-373, 1990.

7. M. Hoehfeld and S. Fahlman, "*Learning with limited numerical precision using the cascade-correlation algorithm*," *IEEE Trans. Neural Networks*, vol.3, No. 4, pp 602-611, July 1992.

8. T. A. Duong and Allen R. Stubberud, "*Convergence Analysis Of Cascade Error Projection-An Efficient Learning Algorithm For Hardware Implementation*", International Journal of Neural System, Vol. 10, No. 3, pp. 199-210, June 2000.

9. T.A. Duong and T. Daud, "*3-D Processor For Real Time Processing To Solve Spatial Temporal Target Recognition*" Accepted to be publish in SCI2001, Florida, July 22-25, 2001.